

# Link

Link A simple and effective action link component for navigation and interactive actions. Features semantic color variants, multiple size options, optional underline styling, and support for custom hex colors with smooth hover transitions. How to use ■ `import { AavaLinkComponent } from "@aava/play-core"` ; Basic Usage ■ Simple link implementations with customizable labels, icons, and styling. Angular Preview Code

```
<!-- Basic Usage --> <aava-link label = " Get Started " color = "
primary " > </ aava-link > <!-- With Icon --> <aava-link label = " Link with Icon " [addIcon] = " true "
arrowDirection = " right " color = " primary " > </ aava-link > <!-- Custom Styles --> <aava-link
label = " Custom Styled Link " [customStyles] = " {'font-weight': 'bold'} " color = " info " > </
aava-link >
```

Colors ■ Semantic color variants and custom hex color support for different contexts. Angular Preview Code

```
<!-- Primary Color --> <aava-link label = " Primary Link " color = " primary "
> </ aava-link > <!-- Success Color --> <aava-link label = " Success Link " color = " success " > </
aava-link > <!-- Warning Color --> <aava-link label = " Warning Link " color = " warning " > </
aava-link > <!-- Danger Color --> <aava-link label = " Danger Link " color = " danger " > </
aava-link > <!-- Info Color --> <aava-link label = " Info Link " color = " info " > </ aava-link > <!--
Custom Hex Color --> <aava-link label = " Custom Link " color = " #7C3AED " > </ aava-link >
```

Available Color Variants ■ Default : Standard link color Primary : Brand primary color Success :

Green for success actions Warning : Orange for caution actions Danger : Red for destructive actions Info : Blue for informational links Custom Hex : Any valid hex color value (e.g., #7C3AED, #14B8A6) Sizes ■ Multiple size options for different layout contexts and visual hierarchy. Angular Preview Code

```
<!-- Small Size --> <aava-link label = " Small Link " size = " sm " color = " primary "
> </ aava-link > <!-- Medium Size (Default) --> <aava-link label = " Medium Link " size = " md "
color = " primary " > </ aava-link > <!-- Large Size --> <aava-link label = " Large Link " size = " lg "
color = " primary " > </ aava-link >
```

Size Options ■ Small : Compact links for dense layouts Medium : Standard size for most use cases (default) Large : Prominent links for primary actions

Underline ■ Optional underline styling for enhanced visual emphasis and accessibility. Angular Preview Code

Features ■ Color System ■ Semantic color variants (primary, success, warning, danger, info) Custom hex color support with automatic validation Consistent hover effects across all color variants Icon Support ■ Optional chevron icons with directional control Left or right arrow positioning Size-responsive icon scaling Customization ■ Custom CSS styles through

customStyles input Flexible sizing options (sm, md, lg) Optional underline styling for accessibility

Accessibility ■ Proper ARIA attributes and role support Keyboard navigation (Enter/Space key support) Screen reader friendly labels Sufficient color contrast for all variants API Reference ■

Inputs ■ Property Type Default Description label string 'Action Link' The text content of the link color 'success' | 'warning' | 'danger' | 'info' | 'default' | 'primary' | string 'default' Color variant or custom hex color size 'sm' | 'md' | 'lg' 'md' Size variant for the link underline boolean false Whether to show underline styling href string " URL for navigation addIcon boolean false Whether to show

an icon with the link `arrowDirection` 'right' | 'left' 'left' Direction of the arrow icon `customStyles` `Record<string, string> {}` Custom CSS styles to apply **Outputs** ■ **Event Type Description** `userClick` `EventEmitter<Event>` Emitted when the link is clicked **Methods** ■ **Method Parameters** **Return** **Type Description** `isHexColor()` `color: string` `boolean` Validates if a color string is a valid hex color `getLinkStyles()` `None` `Record<string, string>` Returns computed styles including custom hex colors `anchorClick()` `event: Event` `void` Handles click events and emits `userClick` event **Properties** ■ **Property Type Description** `separatorIcon` `string` Icon name for the arrow (default: 'chevron-right') `separatorSize` `number` Size of the arrow icon (responsive to link size) `safeHref` `SafeUrl` Sanitized URL for secure navigation **CSS Custom Properties** ■ **Property Default Description** `--link-size-sm-font` Theme-based Font size for small links `--link-size-md-font` Theme-based Font size for medium links `--link-size-lg-font` Theme-based Font size for large links `--link-primary-text` Theme-based Primary color variant `--link-danger-text` Theme-based Danger color variant `--link-success-text` Theme-based Success color variant `--link-warning-text` Theme-based Warning color variant `--link-info-text` Theme-based Info color variant `--link-active-text-decoration` Theme-based Underline decoration style **Best Practices** ■ **Design Guidelines** ■ Use semantic colors to convey meaning and context Reserve large size (lg) for primary actions and important navigation Consider underline for accessibility and emphasis Maintain consistent sizing within related link groups Use custom hex colors sparingly to maintain design consistency Use icons strategically to enhance navigation clarity Choose appropriate arrow direction based on content flow **Accessibility** ■ Ensure sufficient color contrast for all color variants Use descriptive label text that clearly indicates the action or destination Consider underline styling for better visual accessibility Test with screen readers to ensure proper announcement Provide adequate spacing between links for touch interactions **Technical Notes** ■ **Color System** ■ The component supports both predefined semantic colors and custom hex values: Semantic colors use CSS custom properties for theme consistency Custom hex colors are validated using `isHexColor()` method Custom colors are applied directly via inline styles Hover effects maintain consistency across all color variants **Icon System** ■ Icons are conditionally rendered based on `addIcon` property Arrow direction is controlled by `arrowDirection` input Icon sizes automatically scale with link size variants Icons are excluded when underline styling is enabled **Styling Architecture** ■ Uses CSS classes for size and variant styling Combines semantic classes with conditional custom styles Smooth transitions for interactive states Maintains proper text decoration control **ViewEncapsulation** ■ The component uses `ViewEncapsulation.None` to allow global styling: Enables theme-wide link styling consistency Allows CSS custom property inheritance Supports integration with design system tokens

```

<!-- Basic Usage -->
<aava-link label="Get Started" color="primary"></aava-link>

<!-- With Icon -->
<aava-link
  label="Link with Icon"
  [addIcon]="true"
  arrowDirection="right"
  color="primary">
</aava-link>

<!-- Custom Styles -->
<aava-link
  label="Custom Styled Link"
  [customStyles]='{\'font-weight\': \'bold\'}'
  color="info">
</aava-link>

<!-- Primary Color -->
<aava-link label="Primary Link" color="primary"></aava-link>

<!-- Success Color -->
<aava-link label="Success Link" color="success"></aava-link>

<!-- Warning Color -->
<aava-link label="Warning Link" color="warning"></aava-link>

<!-- Danger Color -->
<aava-link label="Danger Link" color="danger"></aava-link>

<!-- Info Color -->
<aava-link label="Info Link" color="info"></aava-link>

<!-- Custom Hex Color -->
<aava-link label="Custom Link" color="#7C3AED"></aava-link>

<!-- Small Size -->
<aava-link label="Small Link" size="sm" color="primary"></aava-link>

<!-- Medium Size (Default) -->
<aava-link label="Medium Link" size="md" color="primary"></aava-link>

<!-- Large Size -->
<aava-link label="Large Link" size="lg" color="primary"></aava-link>

```

■ No code found